Strike Force Asset Allocation using Genetic Search

John McDonnell, Nick Gizzi
Space and Naval Warfare Systems Center
53560 Hull Street
San Diego, CA 92152
mcdonn, gizzi@spawar.navy.mil

Sushil J. Louis
Genetic Algorithm Systems Laboratory
Department of Computer Science
University of Nevada, Reno
sushil@cs.unr.edu

Abstract

We investigate the problem of allocating strike force assets in a dynamic targetting environment using a genetic algorithm. The nonlinear programming formulation developed in this paper encompasses both strike and suppression responsibilities as well as multi-target and multi-threat allocations. Partitioning the allocation strategy matrix into strike and suppression components results in a more effective search. Results on two constructed problems show that the genetic algorithm quickly and reliably finds optimal or near-optimal allocations.

Keywords: Asset Allocation, Genetic Algorithms

1 Introduction

The Real-time Targeting and Retargeting (RTR) program at the Space and Naval Warfare (SPAWAR) Systems Center, San Diego (SSCSD), is developing a Real-time Execution Decision Support (REDS) system that supports real-time retargeting for joint mission strike force operations. The overarching goal of the REDS tools is to provide warfighters with rapid mission planning/replanning, mission execution, targeting and combat assessment capabilities. Utilizing in-theater assets, REDS will allow the warfighter to respond, in real time, to dynamically changing target and threat situations.

The essence of strike force planning consists of allocating a collection of strike assets to a set of targets. An air strike package typically consists of attack, fighter support, suppression of enemy air defenses (SEAD), and C^2 elements. Strike force assets are assigned prior to launch during the mission planning phase which can take up to 12 hours to complete. (Another REDS component, the distributed element level planning tool, compresses this timeline by roughly 75%-85%.) Armed reconnaissance and "kill box" patrolling missions are exceptions to the specificity required for pre-ordained targeting. For these types operations, the mission planners usually select weaponry that is robust against a wide array of targets.

The dynamic nature of the battlefield environment poses a particularly challenging problem for mission strike force operations in which the weaponry load outs of a platform are determined based upon pre-ordained targets and weather forecasts. The present work investigates the capabilities of allocating strike force assets in dynamic battlefield situations. Dynamic reallocation of assets may be warranted when threats "pop-up" during the course of tactical air operations or inclement weather sets in. Other conditions that may also warrant dynamic reallocation of strike force assets include compromised or disabled assets, unexpended ordnance, and the realization of higher priority targets.

A genetic algorithm [Holland, 1975, Goldberg, 1989, Fogel, 1995] is a randomized, parallel search algorthm designed to search poorly understood, nonlinear spaces. As such it seems well suited to this nonlinear problem formulation. Preliminary results reported in this paper provide evidence of this suitability.

The next section describes work related to asset allocation. Section 3 provides a mathematical formulation of the asset allocation problem. This formulation gives the genetic algorithm described in section 4 a fitness function to be maximized. The last section offers results and conclusions.

2 Previous work

Many investigations have been conducted into the optimization of strike force assets. Most, if not all, of these consider the problem of allocating assets to targets prior to launch. For example, Griggs [Griggs et al., 1997] formulated a mixed-integer program (MIP) to allocate platforms and ordnance for each objective. Their MIP is augmented with a decision tree that determines the best plan based upon weather data. Li [Li et al., 1997] converts a nonlinear programming formulation into a MIP problem. Although risk is not explicit in this formulation, it does account for defender suppression. Finally, Yost [Yost, 1995] provides a survey of work conducted that addresses the optimization of strike allocation assets.

A genetic algorithm has also been used for determining optimal aircraft to target allocations. Abrahams [Abrahams et al., 1998] formulates a nonlinear objective function that is used for evaluating allocation strategies. While accounting for effectiveness, joint effectiveness, and risk, their formulation constrains the number of platforms to a single objective. This is unrealistic in light of the fact that a SEAD asset may act to suppress several threats at any given time. Similarly, an attack asset may be expected to attack multiple targets on a sortie.

The strike force asset problem considered here involves the allocation of strike force assets in a dynamic targeting environment. That is, it assumes that weaponry load-outs have been predetermined based upon targeting priorities. Once a strike package is in theatre the enemy's state may have changed, significantly effecting the preplanned mission. Instead of aborting the mission, the strike force assets can be reassigned to new targeting objectives assuming reallocation occurs in a timely manner. As a first step in solving this problem we study the performance and suitability of a genetic algorithm for asset allocation.

3 Problem Formulation

The air-strike package consists of a variety of platforms, each of which has a specific role in the mission. A platform's effectiveness depends on the assets loaded and proficiency of the pilot. The mathematical formulation of the problem is based on the work done by Abrahams [Abrahams et al., 1998]. Modifications were made to accommodate pilot (or smart munitions) performance capabilities and relax the single platform-to-objective constraint that had been imposed.

The goal of the optimizer is to select the most effective platform(s) $P_1, ..., P_m$ to achieve the targeting objectives $T_1, ..., T_n$. The platform allocation strategy is represented as an mxn matrix X where $x_{ij} = 1$ if platform P_i is assigned to objective T_j and 0 otherwise. A platform can be assigned to more than one objective up to a maximum of M_i that a platform P_i can accommodate.

Each platform has a load-out that is configured prior to launch. The total number of assets in the overall strike package is represented as a set of assets $A = A1, ..., Aq$. The binary variable Γ_{ij} represents the assignment of the j_{th} asset to the i_{th} platform. Once the platforms are allocated to targeting objectives, the optimizer assigns the most effective weaponry from the existing load-out to achieve the desired effectiveness. This assignment of resources may be represented as a binary nxq matrix Ω that indicates asset A_j has been allocated to objective T_i.

The proficiency of the pilot of platform P_i in using on-board asset A_j is represented as δ_{ij}. This term can also accommodate the usage of smart munitions that have characteristics independent of a particular pilot's capabilities. If w_{ij} represents the effectiveness (probability) of asset

A_j achieving objective T_i, then, assuming statistical independence between on-board assets in achieving the targeting objectives, the probability of platform P_i achieving objective T_k is

$$p_{ik} = 1 - \prod_{j=1}^{q} (1 - \Gamma_{ij} \delta_{ij} \Omega_{kj} w_{jk})$$

where q is the total number of assets in the strike package.

The overall probability of an objective T_k being achieved by all platforms in the strike package is given by

$$s_k = 1 - \prod_{i=1}^{m} (1 - x_{ik} p_{ik})$$

where m is the number of platforms. We also incorporate a prioroty term, ρ_i, for each objective T_i and included in the objective function as

$$U(X) = \sum_{j=1}^{n} \rho_j s_j$$

where n is the number of objectives.

The coupling of assets to jointly achieve an objective can provide a marginal benfit as described by

$$V(X) = \sum_{j=1}^{n} \sum_{k1=1}^{q} \sum_{k2=1}^{q} \Phi_{jk1k2} \Omega_{jk1} \Omega_{jk2}$$

where Φ_{jk1k2} is the joint effectiveness added by simultaneously assigning assets A_{k1} and A_{k2} to the same objective, T_i. It is important to note that Φ can be negative as well as positive.

The risk to the platforms, as well as their value, should also be addressed in the objective function. If each platform P_i has an associated value v_i, and the risk associated with allocating platform P_i to objective T_j is r_{ij}, then the cost term

$$Y(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} r_{ij} x_{ij}$$

needs to be included in our objective function.

The resulting objective function provides a measure of effectiveness of the strike as well as an evaluation of the risks and costs entailed in carrying out the mission. Combining the effectiveness terms, $U(X)$ and $V(X)$, with the risk/cost component, $Y(X)$, yields the function to be optimized

$$J(X) = \alpha U(X) + \beta V(X) + \gamma Y(X) \qquad (1)$$

$J(X)$ should be maximized in order to maximize the effectiveness while minimizing the risk.

4 Implementation

We used the three-operator genetic algorithm described in Goldberg's book [Goldberg, 1989] but modified the selection operator to use CHC [Eshelman, 1991] selection. Let the population size be N. In CHC selection, the offspring produced by crossover and mutation initially double the population. The new generation consists of the best N individuals from the combined $2N$ parents and offspring.

Our selection strategy induces strong convergence and needs to be balanced by high crossover and mutation rates. We have found that two point crossover with crossover rate of 1.0 coupled with point mutations with a mutation rate of 0.05 works well and these were used in our reported results. A population size of 100 was used and the genetic algorithm ran for 100 iterations.

Each individual was encoded in a binary string composed from substrings corresponding to each platform (see Figure 1). Each substring encodes

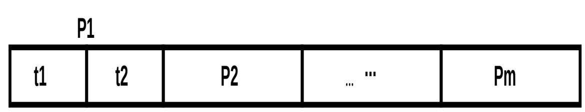

Figure 1: An individual represents an allocation of platforms to targets

the targets that the platform has been allocated and the concatenation of these platform substrings encodes the allocation of platforms to targets for the strike force.

We constructed two test problems to evaluate the genetic algorithm's performance. The problems used fixed values of all platform, pilot, and target properties. Risk was the only factor that varied and we used only the two extreme values of 0.0 and 1.0, with 0.0 risk on the diagonal elements of the risk matrix. This provided us with a known fixed optimum for the problems and thus an objective means by which to evaluate the genetic algorithm's suitability.

The first problem had ten platforms, ten targets, and forty assets. Each platform could be allocated to 2 targets. We need four (4) bits to associate a unique id for each of 10 targets and thus each platform substrings consists of $4x2 = 8$ bits and for 10 targets we end up with a chromosome length of $8x10 = 80$ bits.

The second problem had 20 platforms, 20 targets, and 80 assets. Each platform could again be allocated to 2 targets. We now need 5 bits to associate a unique id for each of the 20 targets and the chromosome length thus ends up to be 200.

5 Results

The graphs in the next section are all averages over ten independent runs of the genetic algorithm with the parameters defined above. Figure 2 shows the maximum, average, and minimum fitness versus the number of iterations of the genetic algorithm. on the first problem while Figure 3 shows the same performance metrics on the second problem.

These graphs indicate that the genetic algorithm easily finds the optimum for these two constructed problems. For a fixed population size and maximum number of iterations, the running time of the genetic algorithm on this allocation problem is dominated by evaluation time (the time needed to calculate the objective function) which grows as the $V(X)$ term in equation 1.

This paper offers a new mathematical formulation of the strike package asset allocation problem and shows that a genetic algorithm offers a viable optimization algoithm for this problem. We would like to study the effect changes in platform, target, asset, and pilot properties, singly and in combination and much work remains to be done.

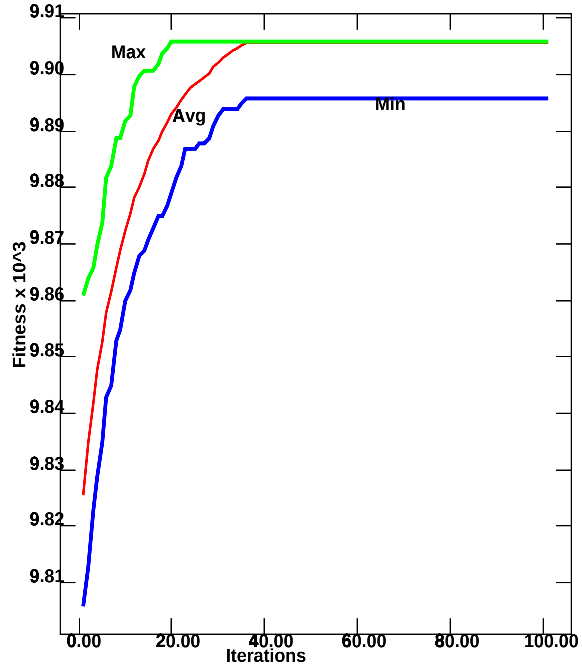

Figure 2: Genetic algorithm maximum, average, and minimum performance on the 10 platform, 10 target, 40 asset problem

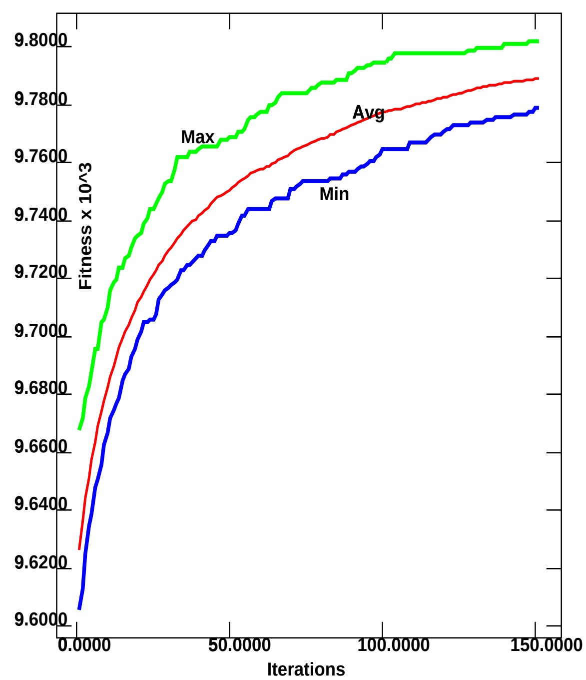

Figure 3: Genetic Algorithm maximum, average, and minimum performance on the 20 platform, 20 target, 80 asset problem

Since the dynamic nature of the targetting environment places rigid time constraints, the next step is to reduce the time needed to produce quality solutions for large allocation problems. We plan to draw on work that combines genetic algorithms with case-based memory [Louis and Johnson, 1997] and test the hypothesis that the combined system offers speed and quality of solution advantages over a genetic algorithm alone.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 9624130.

References

[Abrahams et al., 1998] Abrahams, P., Balart, R., Byrnes, J. S., Cochran, D., Larkin, M. J., Moran, W., Ostheimer, G., and Pollington, A. (1998). Maap: the military aircraft allocation planner. In *Evolutionary Computation Proceedings of the IEEE World Congress on Computational Intelligence*, pages 336–341. IEEE press.

[Eshelman, 1991] Eshelman, L. J. (1991). The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In Rawlins, G. J. E., editor, *Foundations of Genetic Algorithms-1*, pages 265–283. Morgan Kauffman.

[Fogel, 1995] Fogel, D. B. (1995). *Evolutionary Computation, Toward a New Philosophy of Machine Intelligence.* IEEE Press, New York, NY.

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

[Griggs et al., 1997] Griggs, B. J., Parnell, G. S., and Lemkuhl, L. J. (Sep-Oct 1997). An air mission planning algorithm using decision analysis and mixed integer programming. *Operations Research*, 45(5):662–676.

[Holland, 1975] Holland, J. (1975). *Adaptation In Natural and Artificial Systems.* The University of Michigan Press, Ann Arbour.

[Li et al., 1997] Li, V. C.-W., Curry, G. L., and Boyd, E. A. (1997). Strike force allocation with defender suppression. Technical report, Industrial Engineering Department, Texas A&M University.

[Louis and Johnson, 1997] Louis, S. J. and Johnson, J. (1997). Solving similar problems using genetic algorithms and case-based memory. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, pages 283–290. Morgan Kauffman, San Mateo, CA.

[Yost, 1995] Yost, K. A. (Feb 1995). A survey and description of usaf conventional munitions allocation models. Technical report, Office of Aerospace Studies, Kirtland AFB.